# Language and Statistics II

Lecture 6: Log-Linear Models

(Practical Matters)

Noah Smith

# Today’s Plan

- Conditional MLE
- Conditional random fields made simple
- Feature selection
- Regularization

# Log-Linear Models for Prediction

- So far, we've talked about p(*X*), a single random variable.

$$p(x) = \frac{\exp \vec{f}(x) \cdot \vec{\theta}}{\sum_{x'} \exp \vec{f}(x') \cdot \vec{\theta}}$$

- Consider p(*X*, *Y*), where *X* is the **input** and *Y* is the **output**.

$$p(x,y) = \frac{\exp \vec{f}(x,y) \cdot \vec{\theta}}{\sum_{x',y'} \exp \vec{f}(x',y') \cdot \vec{\theta}}$$

# Decoding

- At test time, pick the most probable value of *Y*, given the value of *X*:

$$\hat{y}(x) = \arg\max_y p(x, y) = \arg\max_y p(y|x)p(x) = \arg\max_y p(y|x)$$

- Do we need, then, to model *X*?

# Related

- Recall from last week that we can use log-linear models for language modeling:

$$p\left(W_{i-1} = w \middle| w_1^{i-1}\right) = \frac{\exp \vec{f}\left(w_1^{i-1}, w\right) \cdot \vec{\theta}}{\sum_{w' \in \Sigma} \exp \vec{f}\left(w_1^{i-1}, w'\right) \cdot \vec{\theta}}$$

Denominator depends on history

- I said: "It makes no sense to have features that don't look at the next word at all."

A function that doesn't look at the next word

$$
\begin{aligned}
p\left(W_{i-1} = w \middle| w_1^{i-1}\right) &= \frac{\exp\left(\vec{f}\left(w_1^{i-1}, w\right)\cdot\vec{\theta}\right) e^{g\left(w_1^{i-1}\right)\rho}}{\sum_{w'\in\Sigma}\exp\left(\vec{f}\left(w_1^{i-1}, w'\right)\cdot\vec{\theta}\right) e^{g\left(w_1^{i-1}\right)\rho}} \\
&= \frac{e^{g\left(w_1^{i-1}\right)\rho}\exp\left(\vec{f}\left(w_1^{i-1}, w\right)\cdot\vec{\theta}\right)}{e^{g\left(w_1^{i-1}\right)\rho}\sum_{w'\in\Sigma}\exp\left(\vec{f}\left(w_1^{i-1}, w'\right)\cdot\vec{\theta}\right)} \\
&= \frac{\exp\left(\vec{f}\left(w_1^{i-1}, w\right)\cdot\vec{\theta}\right)}{\sum_{w'\in\Sigma}\exp\left(\vec{f}\left(w_1^{i-1}, w'\right)\cdot\vec{\theta}\right)}
\end{aligned}
$$

# Motivating Conditional Estimation

- Speaking in **general** (not just about log-linear models):

$$p(x,y) = \underbrace{p(y|x)}_{\text{a factor for "}y\text{ with }x\text{"}} \cdot \underbrace{p(x)}_{\text{a factor for just "}x\text{"}} = f_c(x,y)^1 \cdot f_m(x)^1$$

$$p(y|x) = f_c(x,y)^1 \cdot f_m(x)^0$$

# Conditional MLE

- Marginal *p*(*x*) doesn't affect decoding; why bother modeling it?

- Decoding is as before:

$$\hat{y}(x) = \arg\max_y p(x, y) = \arg\max_y p(y|x)\,p(x) = \arg\max_y p(y|x)$$

- **Training** (estimation) is different:

$$\max_{\vec{\theta}} \prod_{i=1}^{D} p_{\vec{\theta}}\left(\tilde{y}_i | \tilde{x}_i\right)$$

# Conditional MLE for Log-Linear Models

**MLE**

$$\max_{\vec{\theta}} \log \prod_{i=1}^{D} \frac{\exp \vec{f}(\tilde{x}_i, \tilde{y}_i) \cdot \vec{\theta}}{Z(\vec{\theta})}$$

$$L(\theta) = \frac{1}{D} \sum_j \theta_j \sum_{i=1}^{D} f_j(\tilde{x}_i, \tilde{y}_i) - \log \underbrace{\sum_{x,y} \exp \sum_j f_j(x,y) \cdot \theta_j}_{Z(\vec{\theta})}$$

$$\frac{\partial L}{\partial \theta_j} = \frac{1}{D} \sum_{i=1}^{D} f_j(\tilde{x}_i, \tilde{y}_i) - \mathbf{E}_{p_{\vec{\theta}}(X,Y)}\left[f_j(X,Y)\right]$$

**CMLE**

$$\max_{\vec{\theta}} \log \prod_{i=1}^{D} \frac{\exp \vec{f}(\tilde{x}_i, \tilde{y}_i) \cdot \vec{\theta}}{Z(\vec{\theta})} \Bigg/ \frac{\sum_y \exp \vec{f}(\tilde{x}_i, y) \cdot \vec{\theta}}{Z(\vec{\theta})}$$

$$L(\theta) = \frac{1}{D} \sum_j \theta_j \sum_{i=1}^{D} f_j(\tilde{x}_i, \tilde{y}_i) - \frac{1}{D} \sum_{i=1}^{D} \log \sum_y \exp \sum_j f_j(\tilde{x}_i, y) \cdot \theta_j$$

$$\frac{\partial L}{\partial \theta_j} = \frac{1}{D} \sum_{i=1}^{D} f_j(\tilde{x}_i, \tilde{y}_i) - \mathbf{E}_{\tilde{p}(X) \cdot p_{\vec{\theta}}(Y|X)}\left[f_j(X,Y)\right]$$

# Is it Still Maximum Entropy?

- Remember, ME(empirical constraints) = MLE(log-linear). What about CMLE?

$$\max_{p} \sum_{x} \tilde{p}(x) H\left(p(Y|x)\right)$$

subject to

$$\forall j, \mathbf{E}_{\tilde{p}(X,Y)}\left[f_j(X,Y)\right] = \mathbf{E}_{\tilde{p}(X)p_{\vec{\theta}}(Y|X)}\left[f_j(X,Y)\right]$$

# Conditional Random Fields Made Simple

- Start with an HMM's features (transitions and emissions)
- All log-probabilities ➔ arbitrary weights.
- Now we have a log-linear model giving *p*(tags, words)
- Train to maximize *p*(tags | words).
  - Required quantities (for $L$ and $\nabla L$) will come from forward-backward algorithms!
- Add more fine-grained features if you want to.

# Maximum Mutual Information Estimation

(Or, the speech people had the same idea!)

$$I(X;Y) = \mathbf{E}\left[\log \frac{p(X,Y)}{p(X)p(Y)}\right]$$

Assume empirical distribution over *X*, *Y*

$$\approx \mathbf{E}_{\tilde{p}(X,Y)}\left[\log \frac{p(X,Y)}{p(X)p(Y)}\right] = \mathbf{E}_{\tilde{p}(X,Y)}\left[\log \frac{p(Y|X)}{p(Y)}\right]$$

Assume *p*(*Y*) is uniform

$$\approx \mathbf{E}_{\tilde{p}(X,Y)}\left[\log p(Y|X)\right] = \frac{1}{D}\sum_{i=1}^{D} \log p(\tilde{y}_i|\tilde{x}_i)$$

# Example

- Suppose we're building a conditional log-linear model over character *j*, given the previous character *j* - 1.

$$f_{342}(c,c') = \begin{cases} 1 & \text{if } c = \texttt{q} \text{ and } c' = \texttt{u} \\ 0 & \text{otherwise} \end{cases}$$

$$f_{343}(c,c') = \begin{cases} 1 & \text{if } c = \texttt{q} \text{ and } c' = \texttt{v} \\ 0 & \text{otherwise} \end{cases}$$

- In training, `q` is **always** followed by `u`. This happens 52 times.

# Example

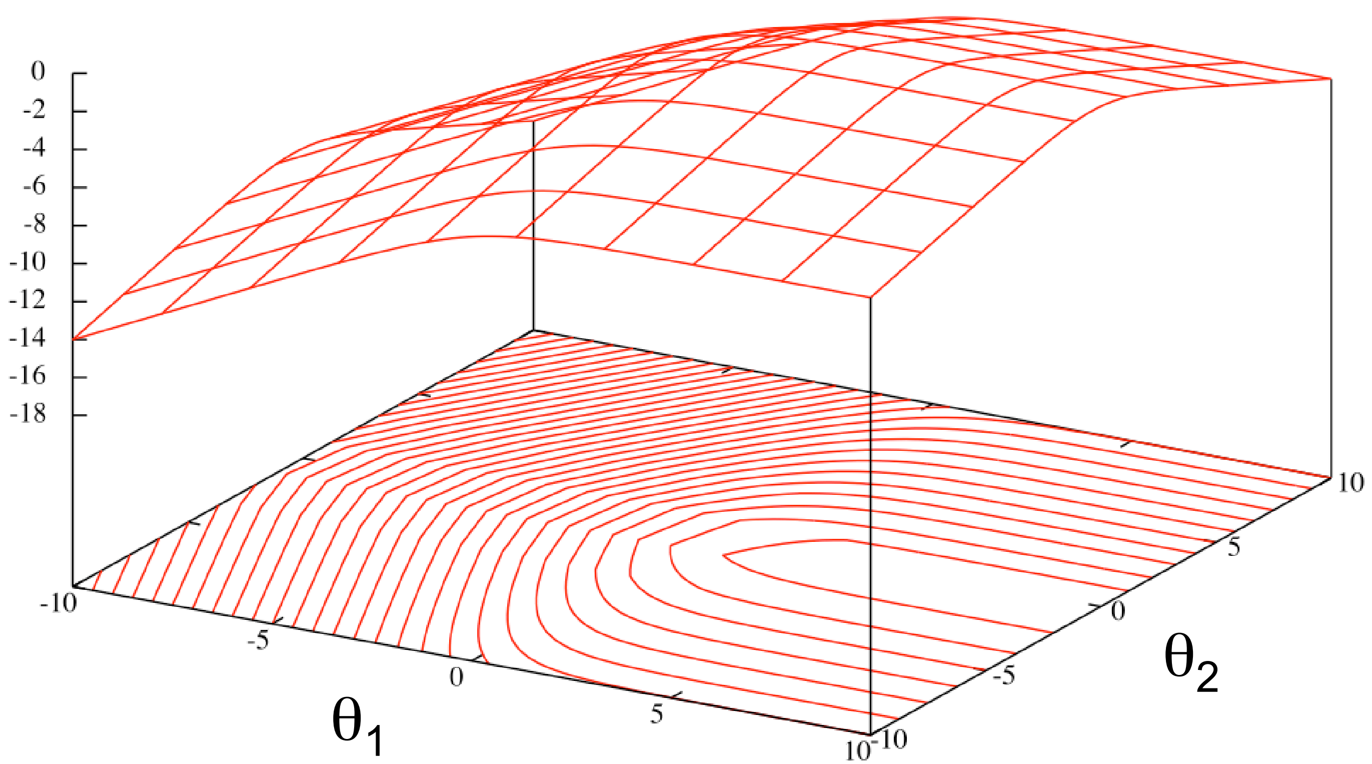
- Ideal for maximizing conditional likelihood: p(u|q) 🡐 1
- To do this, drive $\theta_{342}$ to +∞
- At the same time, drive $\theta_{343}$ to -∞

$$L(\theta) = \frac{1}{D}\sum_j \theta_j \sum_{i=1}^{D} f_j(\tilde{x}_i, \tilde{y}_i) - \frac{1}{D}\sum_{i=1}^{D} \log \sum_y \exp \sum_j f_j(\tilde{x}_i, y) \cdot \theta_j$$

$$\frac{\partial L}{\partial \theta_j} = \frac{1}{D}\sum_{i=1}^{D} f_j(\tilde{x}_i, \tilde{y}_i) - \mathbf{E}_{\tilde{p}(X)\cdot p_{\vec{\theta}}(Y|X)}\left[f_j(X,Y)\right]$$

- Is this really what we want?

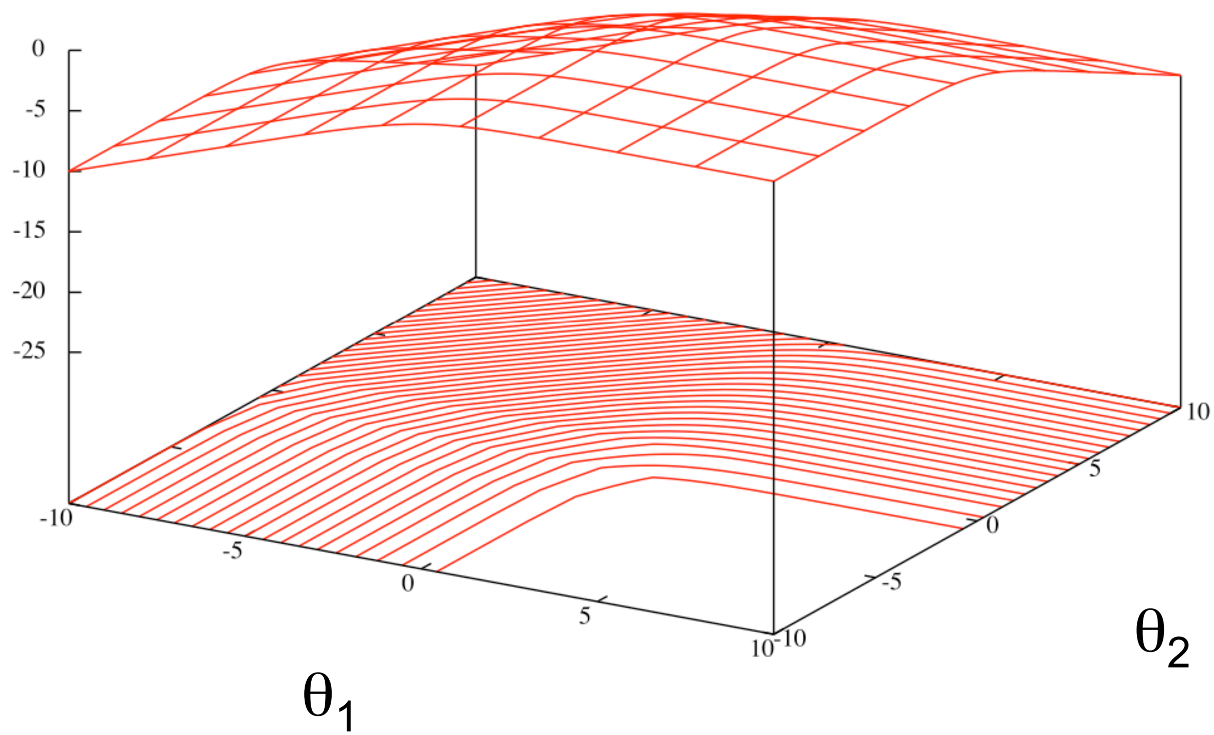
# The infinity problem




$$\mathbf{E}\left[f_1\right] = 1$$

$$\mathbf{E}\left[f_2\right] = 0.4$$

# The infinity problem




$$\mathbf{E}[f_1] = 1$$

$$\mathbf{E}[f_2] = 0$$

# Problems with “Max Ent”

- Training can be expensive
  - Iterative algorithms
  - Inference at each step, possibly involves DP
- No generalization guarantees.
- Based on empirical counts.
- More features ➔ better fit (overfitting).
- Next up:
  - Feature selection
  - Regularization

# Poor Man's Feature Induction (Ratnaparkhi, 1996)

- Include a feature if it is observed five or more times in the training data.

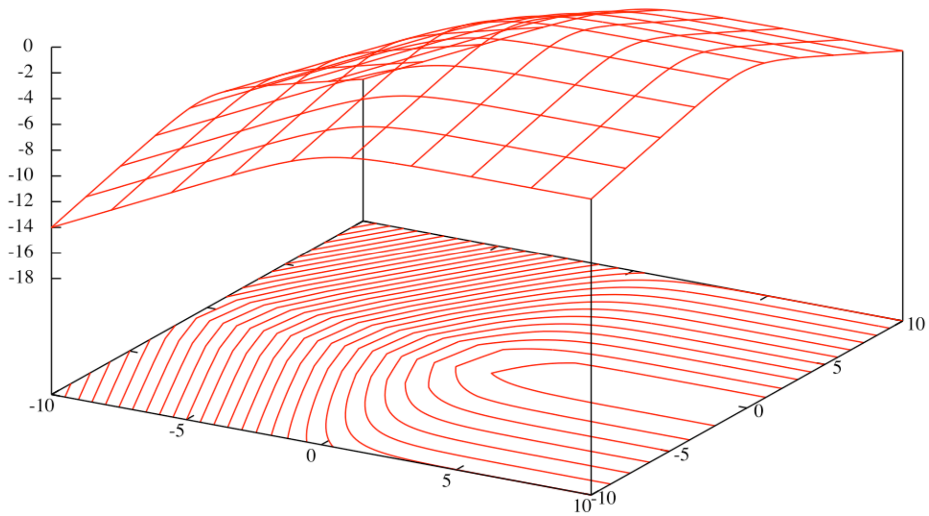
# Feature Induction (Della Pietra et al., 1997)

1. Start with no active features.
2. Consider candidates:
   - “Atomic” features
   - Conjoined features (1 active & 1 atomic)
3. Pick the candidate *g* with the greatest gain.
   - Gain is the maximal improvement over values for *g*’s weight, assuming other feature weights are fixed.
   - Closed form for binary features! (See the paper.)
4. Add *g* to the model.
5. Retrain the model.

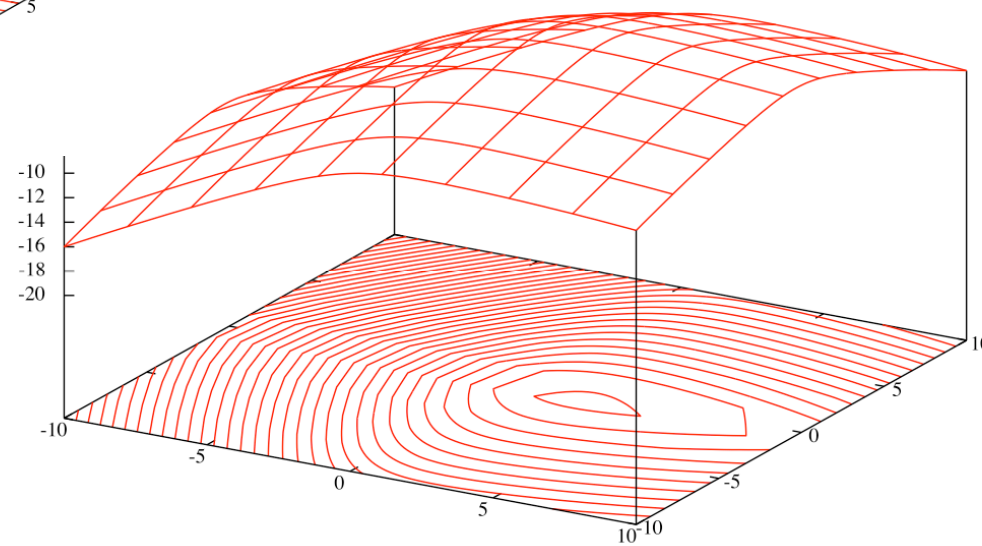
# Regularization

- MLE and CMLE tend to overfit, even for log-linear models.
- Idea borrowed from neural networks: **regularize**, or **penalize** models that are too "extreme."
  - $L_2$: $$\max_{\vec{\theta}} L\left(\vec{\theta}\right) - \underbrace{c\left\|\vec{\theta}\right\|_2^2}_{c\sum_j \theta_j^2}$$
  - $L_1$: $$\max_{\vec{\theta}} L\left(\vec{\theta}\right) - \underbrace{c\left\|\vec{\theta}\right\|_1}_{c\sum_j \left|\theta_j\right|}$$

# $L_2$ Regularization




before




after

# Probabilistic Interpretation

- *Maximum a posteriori* (MAP) estimation:

$$\max_{\vec{\theta}} p_{\vec{\theta}}\left(\tilde{\vec{x}}\right) \cdot p\left(\vec{\theta}\right)$$
$$= \max_{\vec{\theta}} \log p_{\vec{\theta}}\left(\tilde{\vec{x}}\right) + \log p\left(\vec{\theta}\right)$$

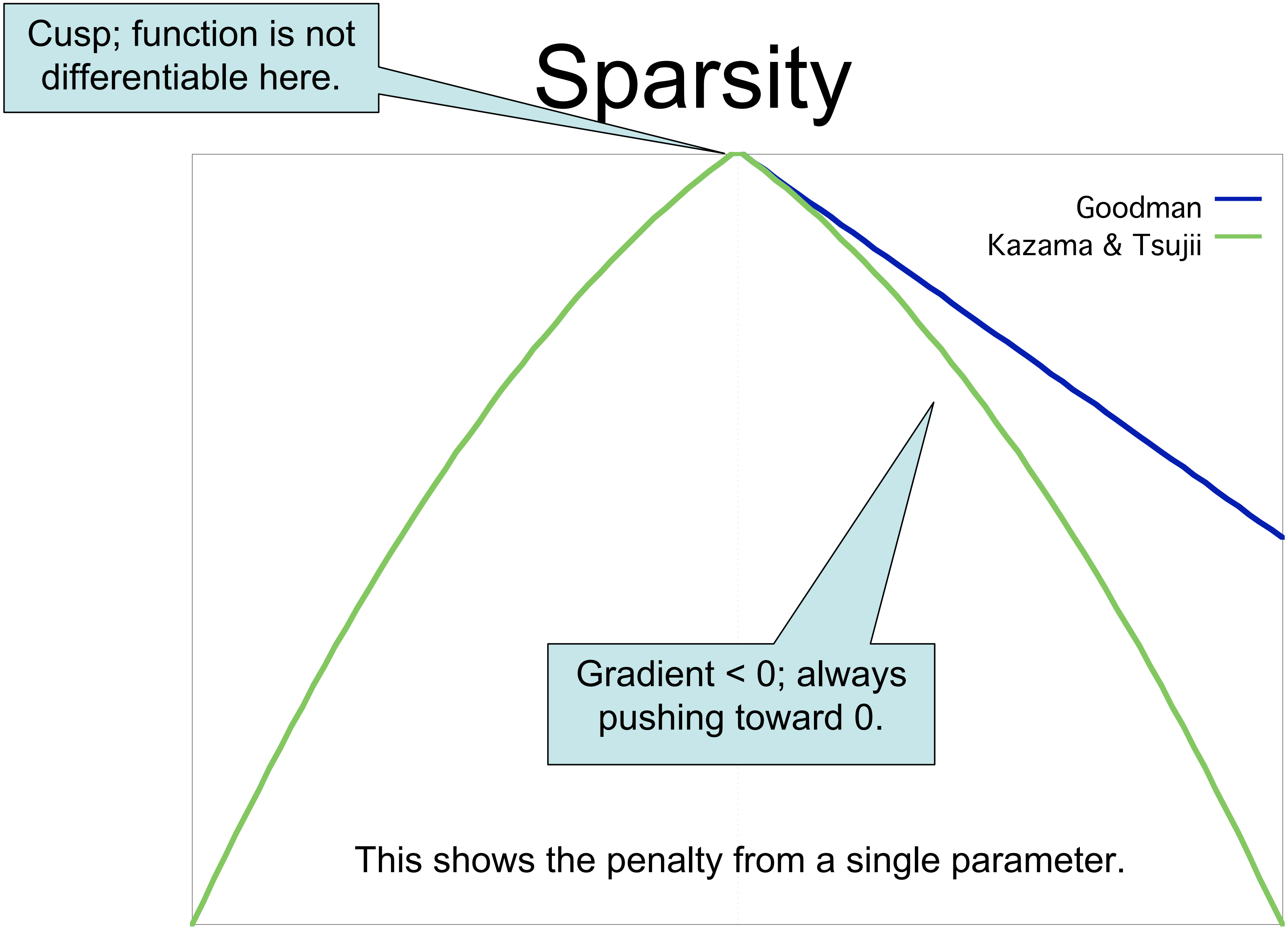
- Zero-mean diagonal Gaussian prior is equivalent to $L_2$ (Chen & Rosenfeld, 1999).

$$\log \mathcal{N}\left(\theta_j; \mu = 0, \sigma^2\right) = const\left(\theta_j\right) - \frac{\theta_j^2}{2\sigma^2}$$

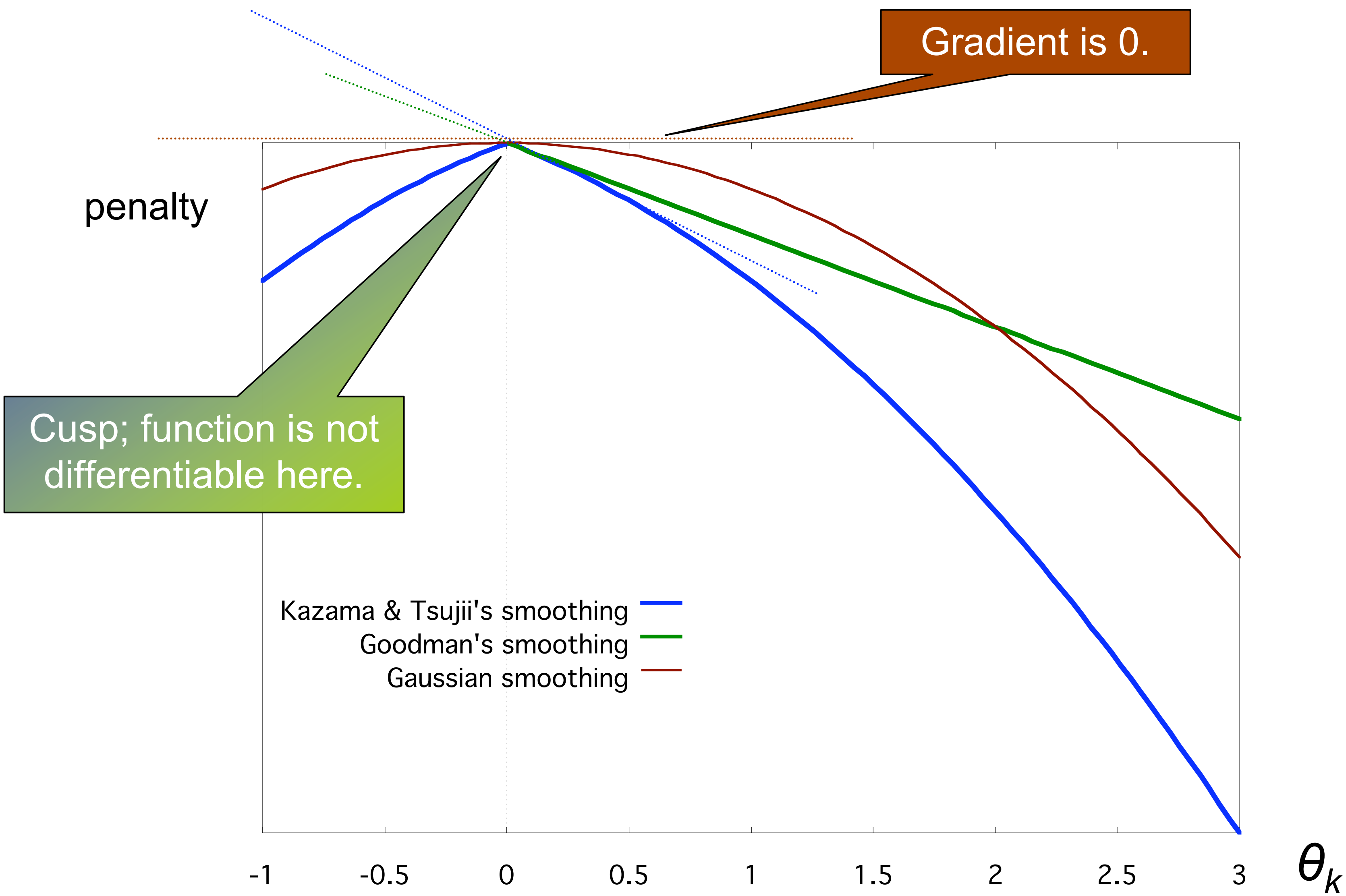
# Probabilistic Interpretation

- Goodman (2003): Laplacian prior corresponds to $L_1$ regularization; also presents exponential prior.
- Related:
  - Kazama & Tsuji'i (2003) and Khudanpur (1995), "relaxed" constraints
- Added bonus for these: sparsity
  - As the prior is strengthened (*c* is increased), more weights go to zero.

# Sparsity


Cusp; function is not differentiable here.
Goodman
Kazama & Tsujii
Gradient < 0; always pushing toward 0.
This shows the penalty from a single parameter.

# Sparsity


Gradient is 0.
penalty
Cusp; function is not differentiable here.
Kazama & Tsujii's smoothing
Goodman's smoothing
Gaussian smoothing
-1
-0.5
0
0.5
1
1.5
2
2.5
3
$\theta_k$

# Wrapping Up Log-Linear Models

- Last Thursday:  the basic idea
  - Features!
  - Informal thoughts about decoding.
- Tuesday:  motivation and training (I)
  - Max Ent and MLE
  - MLE as numerical optimization.
- Today:  training (II)
  - Conditional estimation
  - Feature selection
  - Regularization